**VPMS Lite**

Requirements Specification Document

This Document is prepared with the objective of communicating the requirements that are required by the VPMS Classic lite system.

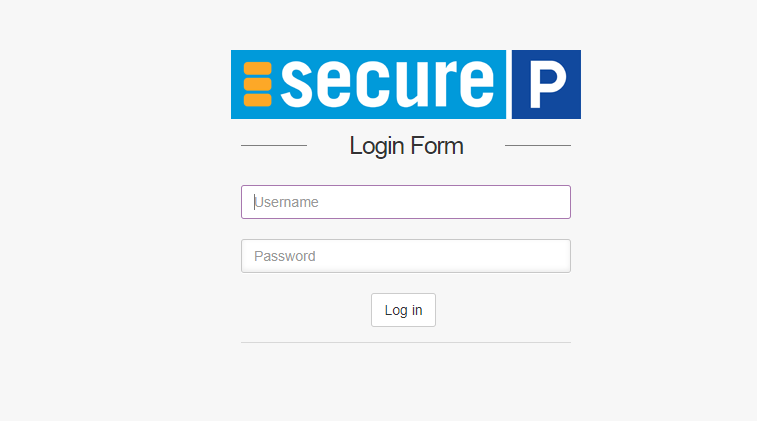
Please NOTE:

1. These requirements are in extension with what features are implemented in VPMS Lite.
2. All the screenshots used are only for reference purpose and should NOT be inferred as the real wireframes / requirements.

Requirements are enlisted screen wise.

Please note: wherever required, a reference to the ADMIN screen will made for the certain settings to be done.

# LOGIN PAGE



| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | Login Page to have a form which will capture User Name & Password |
| 2 | There will be a button called Sign in |

| 3 | There will be an option to click Forgot Password |
| --- | --- |
| 4 | Once this is Forgot Button is clicked, the username OR Email ID should be accepted |
| 5 | On successful input of username or email ID, validate its presence in the list of users & send the password to the Registered Email ID |
| 6 | Show the current date & Time on the top right corner |
| 7 | Show the product name & Secure Parking logo on the top right & left corners respectively |
| 8 | There will be an option to check the box for showing the password |
| 9 | This show password checkbox will decide whether the password being entered on the login page will be displayed or will be displayed as asterisk |
| 10 | Password encryption should be based on Secure Hashes |

1. **CHECKIN SCREEN**

| **Sl.**  **No** | **Requirements** |
| --- | --- |
| 1 | Checkin page shall have the following fields to fill - Ticket No., Veh No., Customer Name, Email ID, Mobile number, Chauffeur and Vehicle type.  There should be drop down to choose the driver name. |
| 2 | Ticket No will be alphanumeric - max up to 15 characters & no spaces or special characters  should be allowed |
| 3 | Vehicle No will be alphanumeric - max up to 15 characters & no spaces or special characters  should be allowed |
| 4 | Driver's name list should be a drop fetching all the active drivers available at that site |
| 5 | Ticket NO to be auto incremented after successful checkin - still editable. Ticket Num to be primary key. Auto increment should be based on the last number that was processed at this  page. Each Terminal to carry out its own ticket no sequence without any conflict to other terminal input |
| 6 | Whenever there is repeated vehicle number, then the NAME, MOBILE NUM and EMAIL ID  should get auto filled. |
| 7 | If there are more than two names / email id / mobile number match is found on one veh number, then there should be a pop up showing all the retrieved info and radio buttons to  choose either of the options. The chosen option should auto fill the fields on the main screen. |
| 8 | There should also be a choice to ignore all the suggestions. In This case, the fields on the main  form will be blank and would need cashier to type in. |
| 9 | After driver’s drop down and vehicle type drop down, there need to be a check box called “Map Parking Ticket”. If this check box is clicked, then a field called “Parking Ticket Number” will become visible and needs to accept inputs with keyboard or through qrcode scanning. This field needs to be alphanumeric only. No Space and No special characters. If the check box is not clicked then this field won’t be visible. |
| 10 | Next, Just like in VPMS-lite there needs to be a list of all the discounts available at that site.  These needs to be displayed |
| 11 | After discount, the additional services should be listed with NAME and PRICE. During checking  in list of additional services can be clicked and availed by the particular vehicle |
| 12 | There can be multiple checkin gates at a particular site |

| 13 | If payment on Entry is configured then, payment details and payment modes needs to  integrated on the checkin page. |
| --- | --- |
| 14 | There needs to be a print checkbox, only if clicked – print needs to be generated. Else direct checkin should happen |
| 15 | At the end of the page Checkin button should be available to complete the process of checking in of the valet vehicle. Since there are multiple checkin gates, there is a possibility that ticket number could be repeated. That’s why ticket num primary key check be present and if ticket is already present, then error response should be shown. In this case, only ticket number field  should become blank, all other fields should be remaining with previously filled data |
| 16 | Next, there should not be a vehicle present already in the valet (meaning checked in but no checked out) if this error is received after checkin button is clicked, then only Vehicle num  field should become blank, all other fields should be remaining with previously filled data |

1. **Controller Screen**

**The below explanation illustrates the flow and what configurations needs to be chosen for a controller screen to be visible**

**PLEASE NOTE: THERE SHALL BE MULTIPLE CONTROLLER GATES at a site if its configured to have controller’s flow.**

In the site configuration screen of any site below should be the flow.

**Car Call** (Radio buttons)

* YES

If “YES” is chosen then there will further one more choice of

**Controller** (Radio buttons)

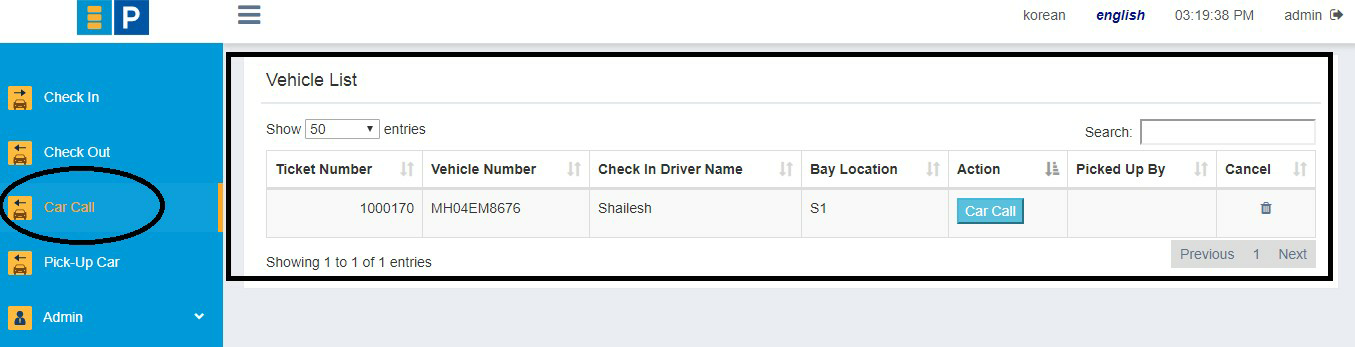
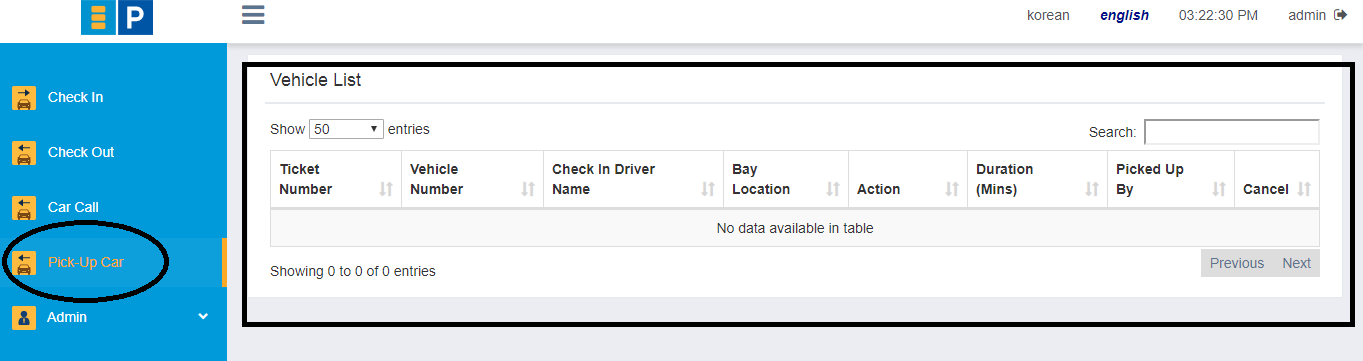
∙ “YES”

If yes is chosen, then there will be a separate screen visible called “Controller”. In this case the “car-call” and the “pick-up car” screens won’t be visible.

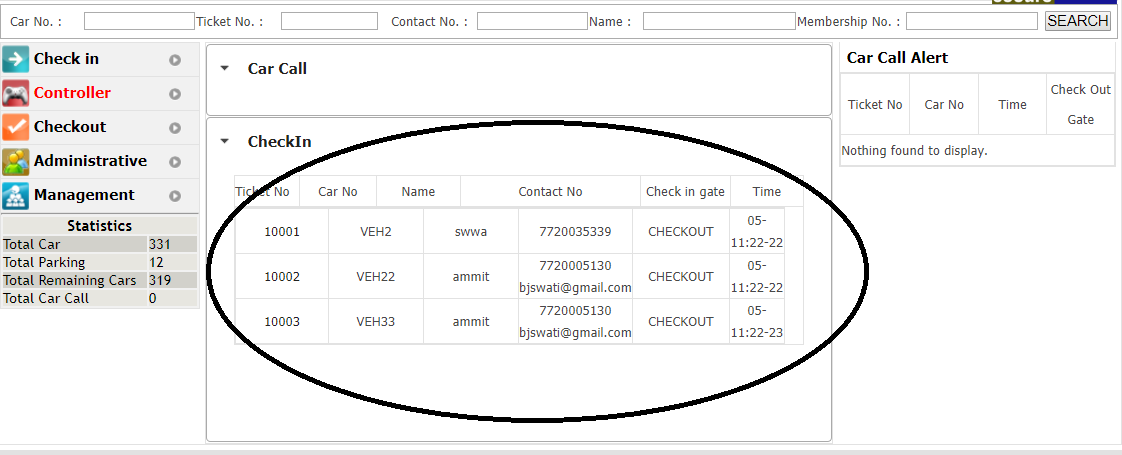
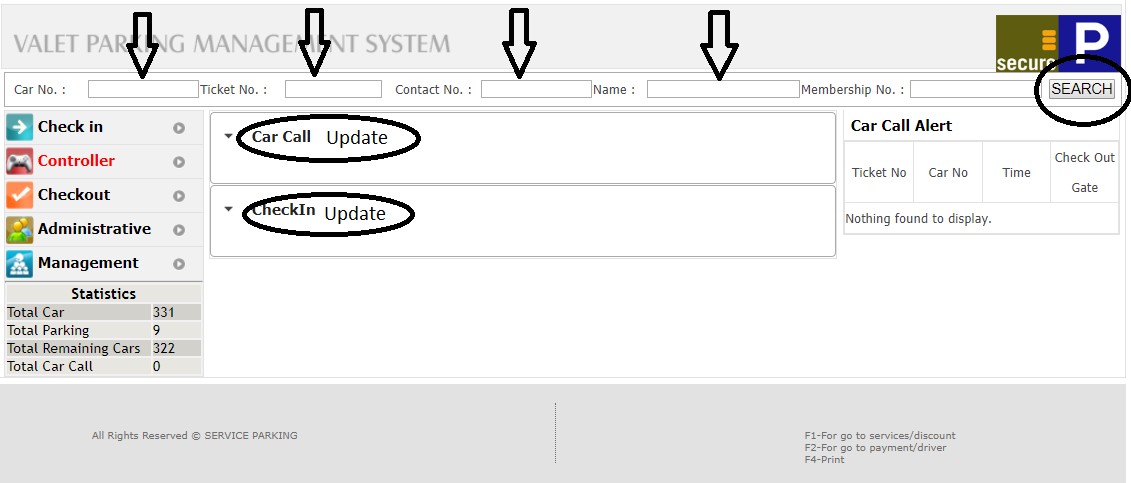
In this case the whole flow of car call & pick up will change as explained below

∙ “NO”

If no is chosen, then it will involve direct car call flow like it is present in VPMS-lite currently ie There shall be only two screens car call and pick up car as shown below

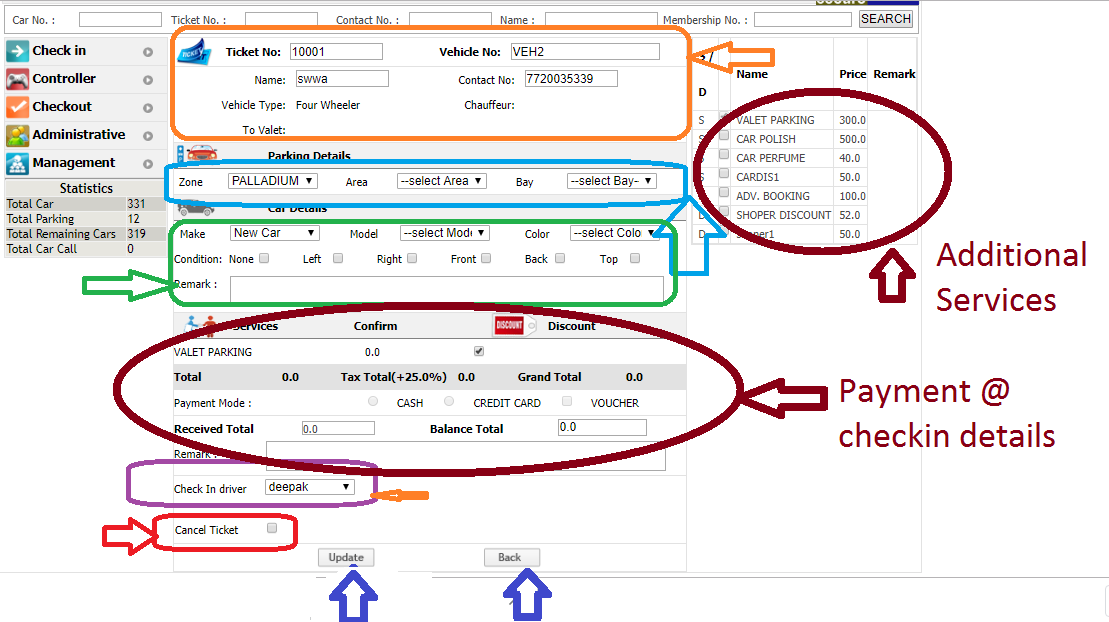
* 1. Car call screen
  2. Pick up car screen
* NO (then simple flow like in VPMS-lite)
  + It will involve only two steps, checkin and checkout.
  + Accordingly, checkin screen will have an extra field called

“Bay Location”



Default view of controller screen

List of checked in tickets visible. Click on the list item shows the below screen



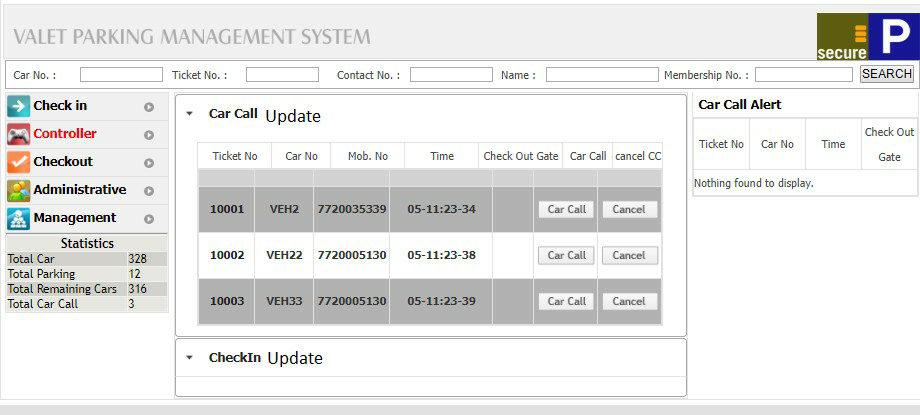
Check in update screen on the controller’s page



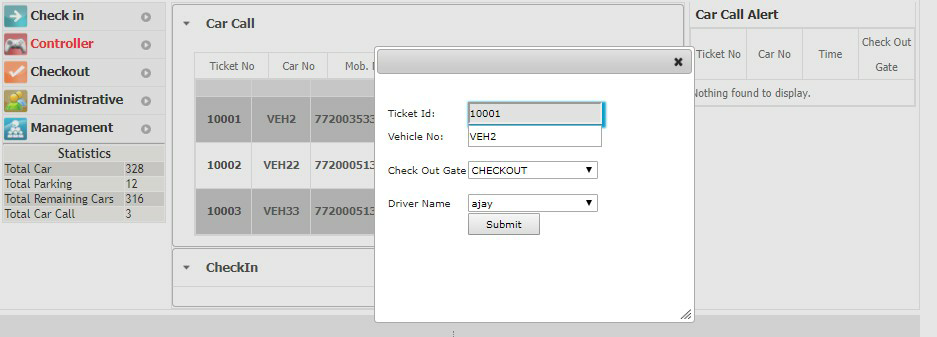
An extra button is visible after the ticket is updated once.

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | Basically, Controller does 3 major tasks.   1. Car call update 2. Checkin update 3. Add additional services |
| 2 | There should be option to search the record based on various fields like VehNum, Ticket No, contact num and name |
| 3 | All the tickets that are checked in are visible under the list of checkin update |
| 4 | Field marked in orange box should show Fields from checkin screen. Few of them should be editable |

| 5 | Field marked in Blue box shows the Zone, area & bay location where the car shall be parked |
| --- | --- |
| 6 | Field s marked in green color shows the make, model and color of the car.  Also it shows the conditions to be marked if any. There is also a remark field to capture additional data |
| 7 | Field marked in purple color shows the check in driver |
| 8 | Field marked in red shows that the controller can do a cancellation of the ticket if checked in by mistake. ( soft delete from db) |
| 9 | Apart from this, the discounts list and the additional services list should also be displayed here. Controller can edit it. These are marked in maroon color circles |
| 10 | If the payment is made at the checkin itself, then payment details also is shown here. In this case payment details, discounts and additional details cannot be edited |
| 11 | There shall be two buttons are the bottom. "Update" and "Back"  Update button click shall save all the changes and back shall come to Controller's default view ignoring all the updates |
| 12 | Once a ticket is updated then it will be no more visible in the list of checkin update. After the update is done once, if a SEARCH is done for this ticket then the same screen will be visible, this time once more button will be visible to perform update and car call. |



Screen shot of Car call update list.



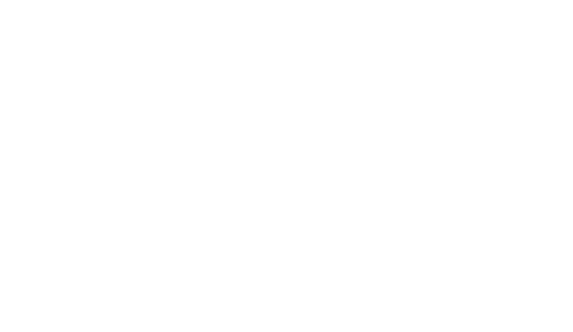
Car call update screen on the Controller’s page.

| 13 | Call call be done in 3 ways   1. through controller screen 2. through checkout screen 3. through the sms link provided to the customers |
| --- | --- |
| 14 | Each of these different ways needs to be colour coded in different ways   1. through controller screen (purple colour) 2. through checkout screen (blue colour) 3. through the sms link provided to the customers (green colour) |
| 15 | After car call update is done, all the text color of these entries shall change to the different colour so that car call update action can be recognized |
| 16 | If the car call is done from the check out gate, then the 3 field will be auto filled with the checkout gate name |
| 17 | if car call is done by the sms link then checkout gate must be filled deliberately. |
| 18 | After any ticket is checked in and the check in update is done by a controller gate, then this gate id will be mapped to this ticket. Hence whenever the car call will be done by checkout/sms/controller, it starts to appear only on the screen on the controller who had done its update. |
| 19 | Car call can also be cancelled. There is a button beside car call update on click of which the ticket moves back to the state of checkin.  While doing this the software should capture the reason and the user logged in. There shall be provision (as option) to charge penalty fee for cancellation of car call |
| 20 | After the car call cancel whenever the controller does a check in update then all the details will be auto filled Except bay location. This shall be the only field that needs to be updated |
| 21 | There needs to be a column which shows the duration after car call but still haven’t checkout. |

State Machine / Flow chart for any ticket when there is a Controller



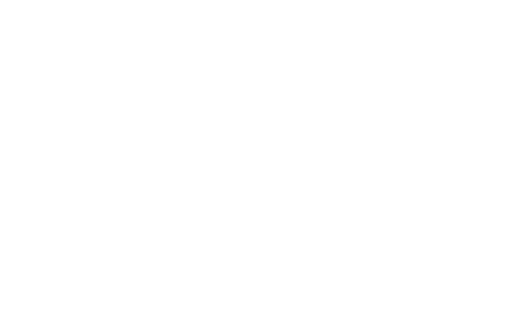
Car call cancel



Step 2 : Check In Update @ Controller

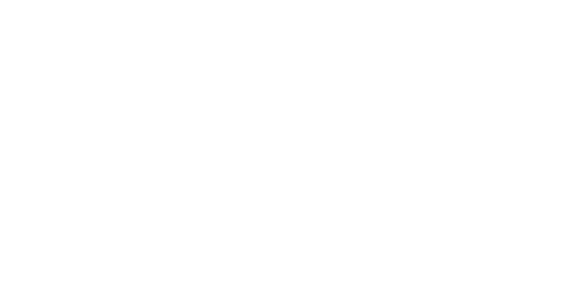
Once the checkin is done, it starts to appear in the checkin update screen

All the fields will be visible on this screen and all the fields from checkin will be editable. Further it shall have more fields to be update



Step 1 : Check In TicketNum, VehNum, Name,

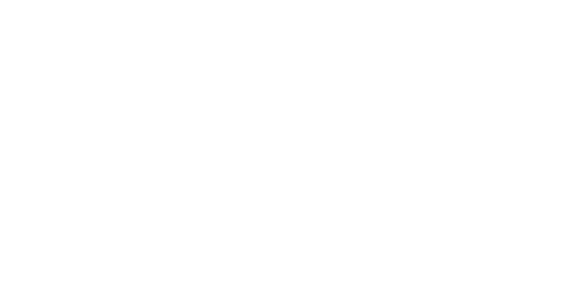
MobileNum, EmailID, Driver, discount, additional services



Step 3 : Car Call

Car call can be done from three places

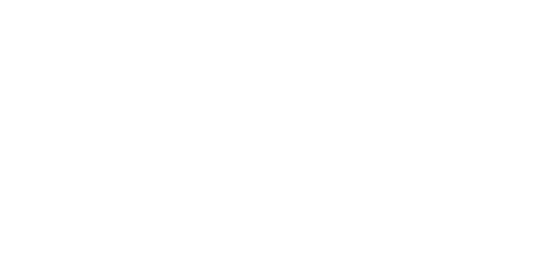
1. Checkout screen
2. By the Customer through SMS link
3. By the controller from controller screen



Step 4 : Car Call Update @ Controller

Once the car call is done it starts to appear in the car call update list.

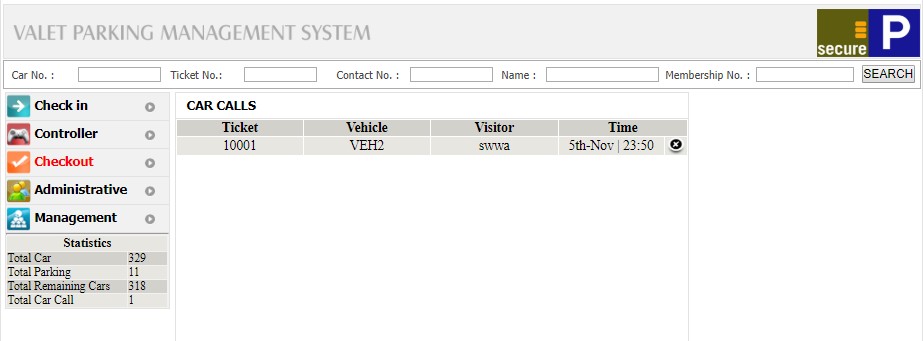
Update the driver who picks up the car and



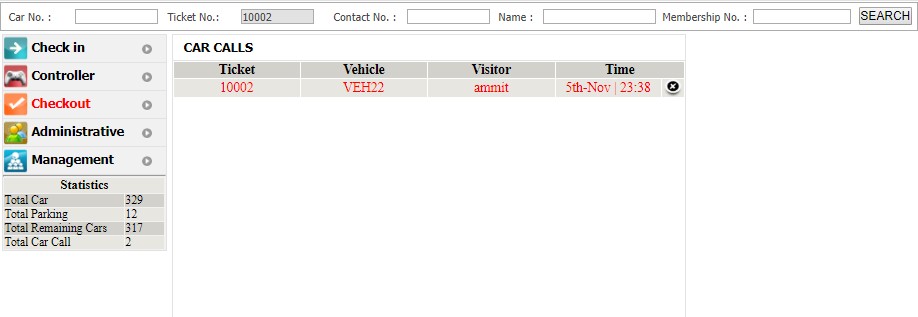
Step 5 : Checkout

Only after step4, checkout of the vehicle can be done

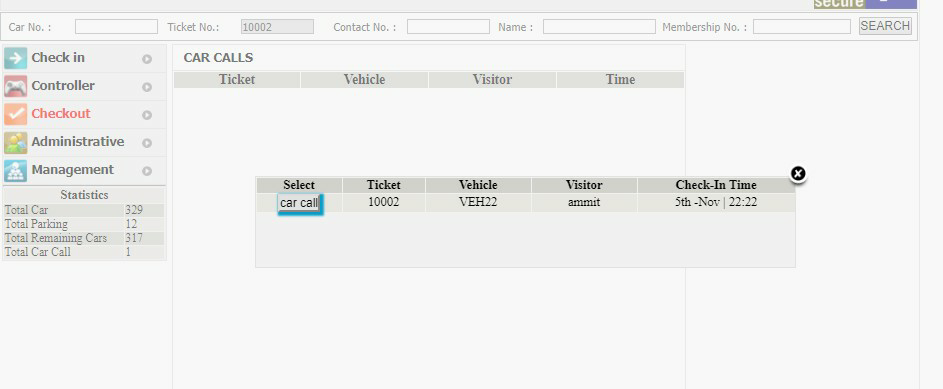
# Checkout Screen



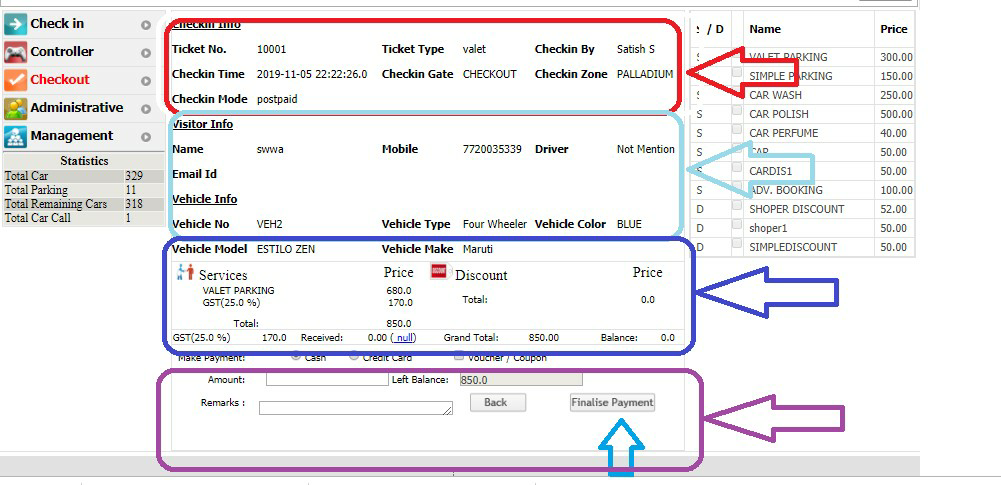
Default view of a checkout screen. This shows the list of all the vehicles to be checked out



The list shows the record in red as long as the car call update is not done by the controller. Once its done, it turns to black like in the above screenshot

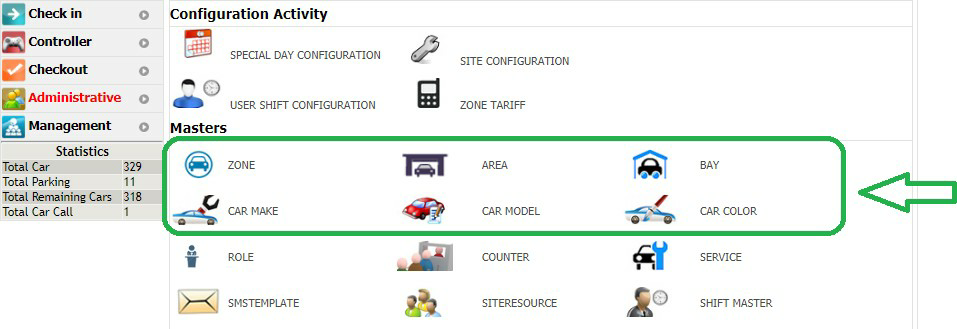


Car call can be done from the checkout screen as well. Ticket num / vehicle num search needs to be done. Then the recognized record is shown like above and click on car call button



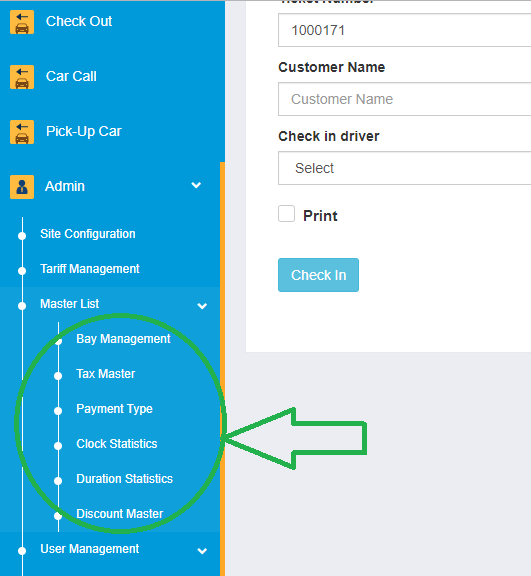
| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | Checkout screen lists the record that can be checked out (meaning all the ones whose car call is done) |
| 2 | It should follow color coding. Display the record in red untill controller does a car call update. Turn it to black only after car call update at controller. |
| 3 | Finalize payment should be available only after the record is displayed in black |
| 4 | After the record is clicked, a list of information is displayed. Refer to the screen shot |
| 5 | 1st set marked in red box should show checkin information ticket num, checkin date & time, checkin gate, checkin user |
| 6 | 2nd set of info in light-blue colour box should be Veh num, name, mobile num, email id, veh type, make, model, color, |
| 7 | 3rd set of info marked in dark-blue box should display list of additional services and discounts applicable. At checkout additional services will not be editable anymore. But discounts needs to  still editable and applicable before finalizing payment |
| 8 | 4th set of info marked in purple color should be details of payment.  It should display the amount calculated , payment mode and a remark's field |

**Please NOTE:**

1. **when there is no car call & no controller, then it should follow the simple flow of checkin and checkout that was initially developed for VPMS-lite**
2. **If there is no controller and only car call is configured then, the flow should be as developed in VPMS lite which shall include car-call and pick-up car screen.**
   1. **Configurations screens**

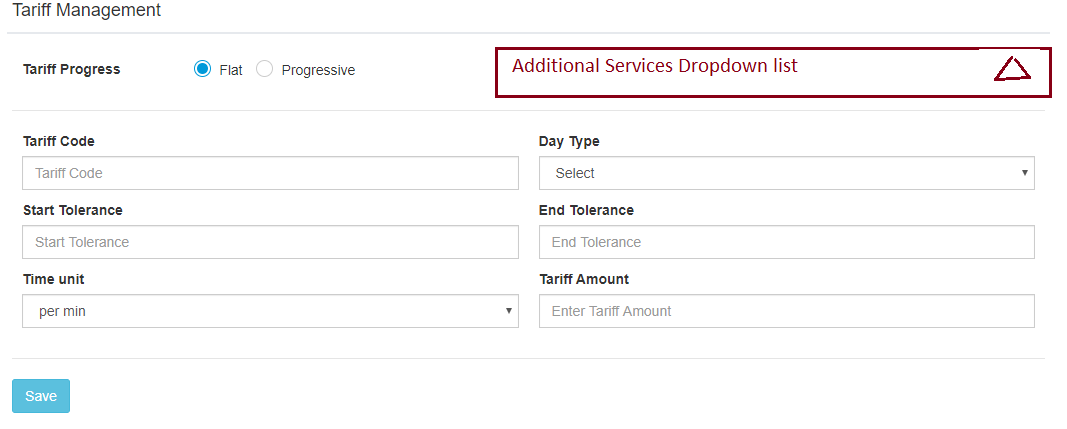
Screen shot which shows options to add car’s make, model and colour. Also Zone, Area and Bay. These should be added under the Master’s list in VPMS lite like shown in the below screenshot Note: 1st Car make needs to be added and then Models will be added mapped to a make

1st zone will be added, then areas will be mapped to zone. Then bays will be mapped to areas



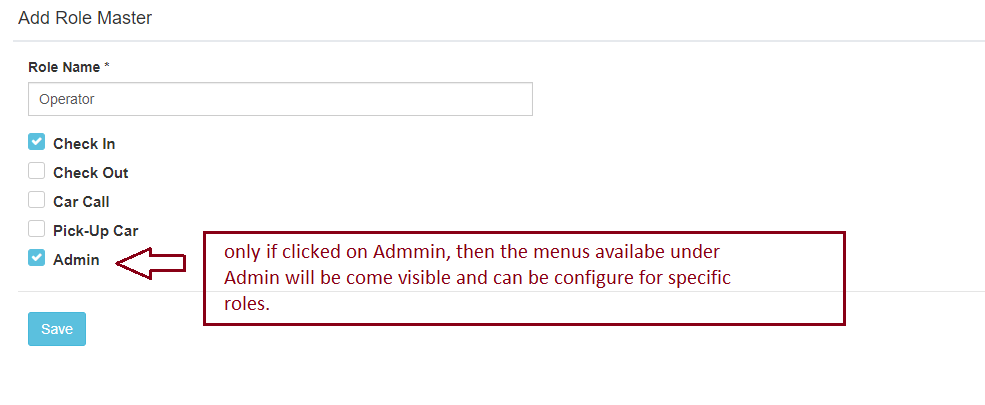
* 1. **Additional Services Menus under Master and Tariff Management**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | There needs to be a menu to add all types of services available at any particular site. For Example, Here I can add the following services 1. Valet Service 2.Car Wash 3. Electric Vehicle(EV) charging Etc |
| 2 | These additional services should be listed as one the parameter while configuring the tariffs. Refer to picture below |
| 3 | Each of these additional services must be visible at the Checking and Controller Screen, So that they can be applied and tariff shall be calculated and the total payable amount can be displayed accordingly |
| 4 | Services like EV charging could be applicable at Pay-On-Entry also, ie at checkin gate. In this case there shall be a flat tariff that will be calculated. |
| 5 | In case of Progressive tariff the calculation should happen at Checkout and the Duration should be considered at the time duration between Controller-Checkin-Update and Controller-Carcall-Update |
| 6 | This logic shall hold good for all the type of services |
| 7 | By default all the tickets/transactions should be applied with Valet Parking Service. This cannot be edited / updated by the controller. |
| 8 |  |

****

* 1. **User Management**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | In addition to adding role and adding users mapped to that role (This is existing behavior), we will need a separate Screen under ADMIN . |
| 2 | This screen will be names as “Operator Management” |
| 3 | This Screen will have 3 sub menus viz. 1. Operator Activation 2. Operator Settlement Input 3. View Settlements |
| 4 | The new requirement is that All the Screens under Admin will also be configurable to be visible / invisible as per the roles defined. Please refer to the below image |
| 5 | So any person who can access the Operator Management is responsible for activating the users for a particular gate. |
| 6 | The screen needs to be designed in such a way that all users with Role = ‘Operator’ will be visible to be mapped with gates. Hence all checkin, Checkout and Controller Gates also should be visible on this screen. It is requested that the wireframe for this screen be shared for approval before starting the development. |
| 7 | All the users will go through the cycle of activation and deactivation based on start shift and End Shift. The cyclic events are as explained below   1. User should be in activated state to start the shift 2. User’s first login after an activation should be considered as “START SHIFT” 3. During the shift, user can login and logged out multiple times. This login logout needs to be recorded in a separate table. 4. Once the users finish the shift, he clicks on the “END SHIFT” Button. Once the user clicks on this button, he becomes deactivated. Any deactivated user cannot login into any gate until he is activated again from the “Operator Activation” Screen 5. All operators who have ended their shift and are deactivated will be visible under “Operator Settlement Input” Screen. Once the settlement is done, the names of the operator will be no more visible on this screen 6. After Settlement is done, all the settlements will be visible in the “View Settlement” Screen. This screen should have a time filter of StartDateTime and EndDateTime, between which all the Settlements would have been done. By default it lists the settlements done during the current calendar’s date. 7. Only after settlement is completed, the user will visible on the “Operator Activation” for the next activation on a specified Gate. 8. If any user has logged in at any gate, then the same user cannot login again at any other gate at the same time. |
| 8 | Please design the screens and share for approval, along with the database schema to capture all the required data |

****

* 1. **Gates Management**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | There needs to be a new and dedicated screen under Admins called “Gates Configurations” where I can add the no. of gates required for that site |
| 2 | This screen should allow to add edit and delete the gates |
| 3 | When I try to add a new gate then there needs to be form which will have the following fields   1. A drop down with these 3 values CI, CO,CN (These will be Short forms for Checkin, Checkout, Controller) 2. A drop which will take in 2 digit numeric number   Note: The combination of these two drop downs will define a unique name for a gate. These names will be saved in the DB Ex: CI01, CO33,CN03, CI11   1. MacID. This should be the ID of the device where the gate should be opened up |
| 4 | Requirements Explanation: At a given site there needs to be 3 Checkin gates, 2 checkout gates, and 3 Controller gates. Then there shall be gates configured like  CI01, CI02, and CI03. ( 3 android devices like Tabs)  CO02 and CO02. ( 2 android devices like Tabs)  CN01, CN02 and CN03. ( 3 desktops) |
| 5 | The url for login page shall be <http://192.168.0.101/vpmslite/public> |
| 6 | So, if this URL is opened up on Android Device1, and if this macID is mapped with CO02, then if any operator whose activation is done for CO02, only this user shall be able to login on this device and no one else. So all transactions happening from this device shall be considered as checkouts from CO02 ie checkout gate no.2 |
| 7 | Also as per the screens configured for this role, only that will be visible to the operator |
| 8 | Flow of transaction will be in such a way that, After a ticket Checkin from any checkin gate, then this will be visible at all the available controller gate for update. Depending on which controller gate updates this record, that gate ID gets mapped to the transaction. Further, if car call is performed then this activity will be visible on the screen of the mapped Controller only. From here only the driver will be assigned and checkout procedure will be performed.  Only exception will be when car call will get cancelled. In such an event, the state of the ticket will go back to checkin->awaiting checkin update. However in this case all the data will be retained and noneditable except the zone, area and Bay of where it will be parked next. |

* 1. **Site Settings**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | The site settings need to have a radio button with YES or NO for sending Exit SMS. |
| 2 | If configured as YES, then there shall be a field where I can be adding the template to send the OTP for the customer. |
| 3 | If configured as NO, then this template will be disabled. And No SMS will be sent to the customer in this case. |
| 4 | If this variable is configured as Yes, then at Checkout there will be 2 step procedures to complete the transactions. |
| 5 | Step1 -> payment finalization  Step2 -> There needs to be an OTP that should be sent to the registered customer’s mobile number. Upon entering this OTP, the transaction closure shall happen |
| 6 | Incase if this is configured as NO, then at checkout it will be only 1 step procedure where the SMS+OTP step will be skipped. |
| 7 | Please note that this feature will be in addition various configurations which are already present in VPMS lite right now. |

* 1. **Additional Reports**

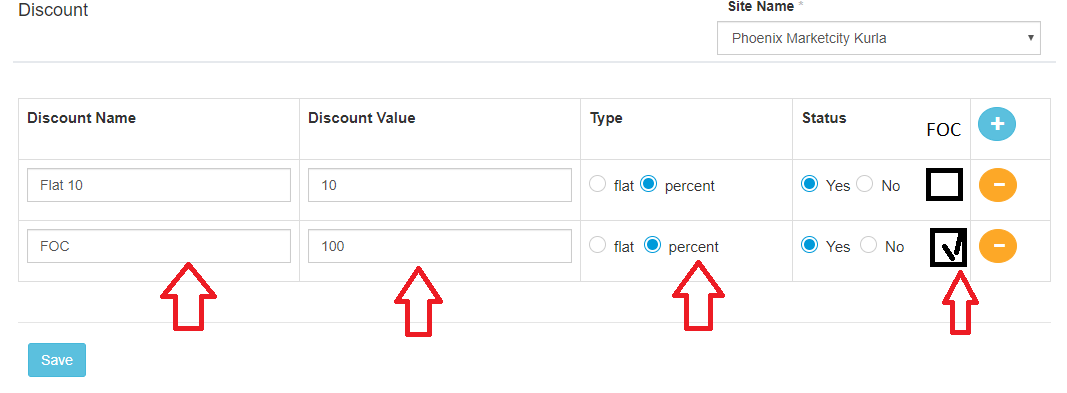
| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | Parking Revenue Data - Covered under Excel Sheet’s Revenue Report |
| 2 | Electric Vehicle Charging Revenue Report- Covered under Excel Sheet’s Revenue Report |
| 3 | Car Wash Revenue Data – Covered under Excel Sheet’s Revenue Report |
| 4 | Hourly Vehicle Activity Report - Covered under Excel Sheet’s Hourly Report |
| 5 | Dwell Time Report – as per the current format of VPMS Lite |
| 6 | Driver Efficiency Report and Driver travel time report - Covered under Excel Sheet’s Driver Report |
| 7 | Customer Data Report - Covered under Excel Sheet’s Customer’s Data Report |
| 8 | Customer Frequency Report - Covered under Excel Sheet’s Frequency Report |
| 9 | Operator Efficiency Report –This report needs to capture and show the session or the time taken by every operator to perform the designated activity |



* 1. **Special Discount for FOC (Free Of Cost)**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | While adding discounts from the discounts screen, there needs to be an extra checkbox called FOC. Refer the Picture below |
| 2 | If this checkbox is checked, then it will be 100% discount and the total payable amount for the customer will be zero. |
| 3 | However, While doing settlement of the cashier who would have used FOC discount, then there will be a special flag that will be raised for the transaction. |
| 4 | The transactions with this FOC\_flag =1 will be visible during the settlement input and the supervisor/ manager / admin whoever is doing the settlement should approve these FOC transactions. |
| 5 | If any of the FOC transactions are not approved in the settlement input screen then this amount should be added for recovery. |
| 6 | If the recovery is not done then the settlement procedure will not be completed. |
| 7 | The settlement Logic is explained in the below attached excel sheet below. |



****

**Some Points to be noted:**

* + 1. We intend to use the same db that’s currently used for VPMS lite. Obviously, we understand there shall be structural changes in db. But idea is to have single db for all types of configurations
* Checkin -> checkout
* Checkin -> car call -> pick up car -> checkout
* Checkin -> Controller : checkin update -> car call -> Controller : car call update -> checkout
  + 1. Site configurations shall remain same as that of VPMS lite. Additional things required for VPM classic implementation. Ex: car call and controller
    2. Tariff and taxes printing logic shall remain same. Whether paid at entry (prepaid) or at checkout (postpaid) printing checkbox will decide whether receipt / ticket to be generated or no.

Receipt layout and details to be printed – this logic shall remain same. The receipt layout will comply to GST invoice requirement.

* + 1. Checkin SMS template and Checkout SMS template shall also remain as it is. However whether the sms link to do car call should be send or no should be a configurable option. If ticked yes, then a field become editable where a message will be typed. This message along with sms will get append to the checkin sms

Ex : “checkin-message”+ “sms message” + “link”

* + 1. Whatsapp message to be integrated with the system to send ticket and receipt to customer phone number. SMS message will have the link to download the GST invoice.
    2. There is a need to integrate two different payment modes.
* EDC Machine
* Paytm

Details of integrating this will be separate document.

* + 1. The cashier’s activation and settlement logic should remain as it is from VPMS classic
* First input the cash and then system displays if the amount is less / more or equal.
* If less, then recovery needs to be done
* Only after recovery / settlement cashier will be available for next shift. Once end shift, only after settlement he / she can be visible.
  + 1. There needs to be a tab which shows about the current statistics

Total no. of Bay = 100 No. of bays used = 8

No. of bays remaining = 92

No . of car calls done = 12 No. of cars picked up = 8

* + 1. For Reports, we will need all the reports currently present in VPMS classic along with lite version
    2. Additionally, we will need a new report called “car call report” which shall show the details like
* How was car call done
* What time stamp car call was done
* When was car picked up
* Duration between car call and checkout
  1. **QR code authorization & Driver lifecycle**

Note: Drivers are the most important roles in the operations of Valet Parking. They have two main jobs

* + 1. Pickup the car from the customer and park it -> drop the key at the controller desk
    2. Collect the keys from Controller desk -> drive the car from parking lot to drop off area and hand over the keys to customer

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | Every driver shall have a wristband, on which a QR code will be printed. The driver shall scan the same at multiple points whenever required. |
| 2 | This is a reference to the User Management ie #7. Whenever any user is added mapped to the role of driver, then there shall be a unique QR code string that will be generated for him/ her.  This should be unique mandatorily – could be a combination of firstname\_lastname\_mobilenum |
| 3 | This QR code should be shown on the screen as a thumbnail and needs to have an option of viewing it in a large view when clicked. This feature is required to print the same on the driver’s wristband. |
| 4 | Once a driver is assigned for a ticket at checkin, then he / she must be blocked and should not be seen in the drop down at any other place like – other checkin gates, checkout gates and controller desks. This action is called “BLOCKING OF DRIVER” |
| 5 | Next action shall be “UNBLOCKING OF DRIVER”. This should happen when the driver finishes the job and reports. After this action, the driver will become visible again at all gates to take up the next job. |
| 6 | Summarizing the flow / sequence:  **Flow 1: Checkin gate to parking :**  Assign driver at checkin gate -> “Block the driver” -> driver drives the car & parks -> driver drops the key at controller desk-> Controller “Unblocks the driver” -> Driver is now free to take up next job  **Flow 2: Parking to checkout gate:**  Controller assigns the driver to a ticket -> “Block the driver” -> driver drives the car to the checkout gate -> after the car is handed over to customer & payment is done -> finalization of checkout -> “Unblocks the driver” -> Driver is now free to take up next job |
| 7 | Assigning a driver to do the task could be done in either ways, select from the drop down OR scan the QR code to recognize the driver.  This requires the application to be equipped with a scanner to read the input.   * If the application is opened on desktop, then a physical scanner will be attached and QR must be read from there. * If the application is on opened up on mobile, the apk should be capable of opening the camera to scan the QR code. |

* 1. **Digital Key Rack Management**

Note: The hardware system to manage the keys, locking and unlocking of the key hooks shall be managed by a separate hardware vendor. There shall be a few API integrations that needs to be done to integrate the VPMS system to the Digital Key Rack Management system (DKRMS). Below will be the scenario / sequence diagram to integrate the DKRMS.

The below APIs are only for illustrations and understanding purpose. The actual scope of integrate shall differ as per the hardware specification

13.1 This integration should happen only on the Controller Desk / System

13.2 The physical location of the DKRMS shall be in the controller’s room where the keys are managed by the operations team.

13.3 The operating of the DKRMS should be available only to the users who login as the controller’s role.

2

API call to get “Assign a open a lock – send arguments like hook number & Ticket num”

API response - The lock opens and a LED blinks for the requested ticket num

The VPMS system should save the response. Controller picks the keys and clicks on proceed

Acknowledgement to call next API – job done, so that hook can get closed again

1

API call to get “Assign a lock to ticket and open a locked-hook”

API response with hook number. The lock opens and a LED blinks for that hook

The VPMS system should save the response and lock-hook number against the ticket / Veh num. Controller places the key and acknowledges.

Acknowledgement to call next API – job done, so that hook can get closed again

VPMS

DKRMS (digital key rack management system)

* 1. **Performance Management**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | This requirement is about capturing details of all the role’s users and their respective time stamps, so that the performances could be evaluated. Below are the scenarios mentioned where the details needs to be captured and saved in the db. |
| 2 | Checkin: 2.1 start shift & end shift time stamp of the Cashier who logged in at a gate.  2.2 cashier’s name to be noted for all the tags he enters into the system  2.3 time stamp of every tag that will be checked in to the system  2.4 time stamp and driver’s name who will pick up the car |
| 3 | Controller: 3.1 start shift and end shift time stamp of the user who logs in as controller at the controller desk   * 1. Every time stamp that the controller user uses the DKRMS, for which activity – lock / unlock and for which tag / vehicle number and all the tickets that the controller updates needs to be captured in db (name & time stamp)   2. The time stamp of when the driver finishes the car parking and drops keys to the controller also need to be noted   3. Every time, whenever the driver is assigned to pick up the car from the parking to drop it back at the checkout gate where the customer will be waiting to take the car back. This time stamp also needs to be noted. (name & time stamp – against the ticket) |
| 4 | Checkout Gate:  4.1 Start shift and End shift timing of the cashier who would be logged in into the checkout gate (user’s name and timestamp)  4.2 the time stamp and the username of the driver who gets the car of the customer and hands over at the checkout desk  4.3 The time stamp of the checkout done for the given ticket  4.4 time stamp of the payment done / finalization of the payment |
| 5 | Car Call:  5.1 The time stamp of car call done needs to be captured  5.2 If the car call is done from any gate, then the gate name and user who did the car call also needs to be captured  5.3 If the customers do a call by clicking on the link sent to them by sms/whatsapp/email, then it’s time stamp and the flag also should be noted that customer did a car call |
| 6 | Every time stamp, user, gate and activity needs to be captured so that this data shall be used to create the reports |
| 7 | The details of the performance related reports is illustrated under the reports section |

* 1. **Partial Payment Module**

This section is related to the tariff which is detailed in Section #6 ie Tariff Management.

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | If the existing tariff module is observed, then there are two types of tariffs. 1st is Flat and 2nd is progressive |
| 2 | **Site Settings Page:** In the settings page, there is radio button which defines where the tariff needs to be collected. Either at the checkin gate or at the checkout gate. We need third button here which says “Partial Pay at Entry” |
| 3 | If this option is chosen, then the tariff that is configured needs to be applied in the way that is explained below. |
| 4 | The first tier should always be “Flat” and this amount needs to be collected and applied at the checkin gate itself. The following next set of tiers could be either flat or progressive. |
| 5 | If the customer denies to pay at the entry and wishes to make a full payment while checkout, then the Cashier at the exit should mark that as FOC. The details of this behavior is explained in section #11 |
| 6 | Explanation with example of this feature.  Tariff – Progressive  1st tier “Flat” 3hrs 500Rs  2nd tier “prog” 1hr 50Rs |
| 7 | Use Case 1: Checkin time 4pm  Checkout time 10pm  PayOnEntry (POE) = Yes  Calculation Logic –  @ Checkin Gate it’s a straight forward logic that the 1st flat tier needs to be applied.  @ Checkout Gate Full tariff needs to get calculated first ie 4pm to 10pm will be 6 hrs  1st tier applied – 3 hrs – 500rs  2nd tier applied – 1hr - 50rs  2nd tier applied – 1hr - 50rs  2nd tier applied – 1hr - 50rs  ----------------------------------------------  Total Calculated - 6hrs - 650rs  ----------------------------------------------  (-) Amount Paid at Entry - 500Rs  Hence Total Payable at Checkout Gate will be (650-500) = 150Rs |
| 8 | Use Case 2: Checkin time 4pm  Checkout time 10pm  PayOnEntry (POE) = NO (FOC ie Free of Cost was applied by the cashier)  Calculation Logic –  @ Checkin Gate it’s a straight forward logic that the 1st flat tier needs to be applied.  @ Checkout Gate Full tariff needs to get calculated first ie 4pm to 10pm will be 6 hrs  1st tier applied – 3 hrs – 500rs  2nd tier applied – 1hr - 50rs  2nd tier applied – 1hr - 50rs  2nd tier applied – 1hr - 50rs  ----------------------------------------------  Total Calculated - 6hrs - 650rs  ----------------------------------------------  (-) Amount Paid at Entry - 0Rs -> Because FOC was done and zero rs was paid  Hence Total Payable at Checkout Gate will be (650-0) = 650Rs |
| 9 | Use Case 3: Checkin time 4pm  Checkout time 7pm  PayOnEntry (POE) = YES  Calculation Logic –  @ Checkin Gate it’s a straight forward logic that the 1st flat tier needs to be applied.  @ Checkout Gate Full tariff needs to get calculated first ie 4pm to 10pm will be 3 hrs  1st tier applied – 3 hrs – 500rs  ----------------------------------------------  Total Calculated - 6hrs - 500rs  ----------------------------------------------  (-) Amount Paid at Entry - 500Rs  Hence Total Payable at Checkout Gate will be (500-500) = 0Rs |

* 1. **WhatsApp & Email Triggers**

This module is about sending the emails and Whatsapp triggers

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | With Reference to Section #9, where we have an option to enable / disable SMS,  Similarly we need to have radio button options for Whatsapp and Emails too |
| 2 | If the radio buttons are enabled, then the whatsapp message needs to be sent to the registered mobile number. |
| 3 | Similarly if enabled to send emails, then email needs to be sent to the registered email address |
| 4 | The template for Whatsapp needs to be defined separately. Like we had for SMS, there shall be a separate text field to configure the message to be sent on whatsapp also. It should contain certain variables that needs to be picked up from code / real time data. |
| 5 | Similar separate template should be available for configuring Emails too. |
| 6 | There are three purposes to send SMS, WhatsApp and Emails to the customers   1. Send a confirmation message on successful Checkin and parking of Car   This shall also have a link through which the customer can do a car call   1. A OTP will be sent after car call which needs to be shared with the cashier / driver at the exit 2. The paid receipt will be sent on Email |
| 7 | We need to have two different templates to configure for Emails.  1st – for Sending OTP after car call  2nd – for sending the Paid receipts. |

* 1. **Mobile APK + POS integration**

| **Sl. No** | **Requirements** |
| --- | --- |
| 1 | There needs to be an Android APK which needs to be compiled to be running on two devices   * Any Android Mobile * A integrated POS where there is inbuilt printer and EDC swipe options to make card payments |
| 2 | In the database, please save the different values to differentiate the different payment modes like please save the different values to differentiate the different payment modes like   * Cash * EDC * UPI |
| 3 | Printer integration:   1. If the APK is on android mobile, then there will be an EPSON printer with whom it should make a BT connection and print the tickets/ paid receipts 2. If the application is running on the POS, then there will be an inbuilt printer whose library API calls needs to be done to print the tickets/ paid receipts |
| 4 | UPI Payments: There will be dynamic QR code that will be generated for Every customer.   * On android mobile, there must integration needs to be done with the payment banks like Paytms / others * On POS, the UPI will be done with the POS providers like mswipe / paytm |
| 5 | EDC : on the POS there will be swipe / insert slots which needs to be integrated with our APK.  This will be achieved by calling the APIs from the inbuilt library available on the POS. |
| 6 | This APK should render the actual VPMS lite web pages. It should be progressive web app. |